**Q1) Write the following code and answer the following questions.**

a) After compiling the above code, how many .class files are generated?

=>

Two Class files are generated.

Because JVM is not aware about inner classes because compiler convert inner class into regular class files(for example, OuterclassName$InnerclassName.class).

b) List the name of the .Class files

=>

1) Uiversity.class

스크린샷이(가) 표시된 사진

자동 생성된 설명2) University$Department.class

**Q2) Write the following code and answer the following questions.**

a) When you compile and run the above program, you will get an error? Why this error is generated?

=> Yes. Because as you can see inner class is non-static, but the main method (which is static) is in the inner class. static methods can only be declared in a static type.

b) Cannot a non-static inner class have a static method including main()?

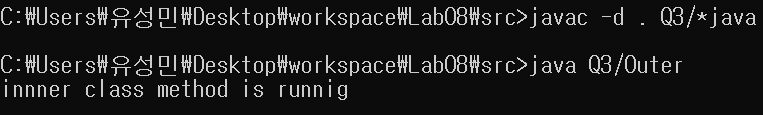
=> No. Because inner class is non-static, so it can’t access static method. To access the static method we should make outer class’s object and then we can use static method with outer class.

텍스트이(가) 표시된 사진

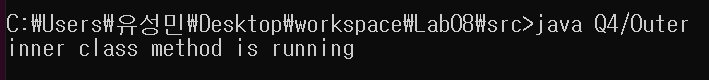
자동 생성된 설명

**Q3) The following code is a partial code. Complete the missing code to get the output.**

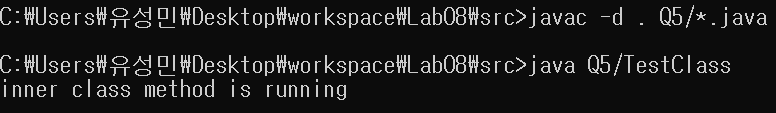
=> To access from static main method, we can’t access non-static directly. But after making object of the inner class, we can access non-static M1() method. And when we make inner class’s object it is important to write outer class’s instance before ‘new’. So as you can see, I made Outer class’s object and Inner class’s object and then accessed M1() method at static main method.



**Q4) The following code is a patial code. Complete the missing code to get the output.**

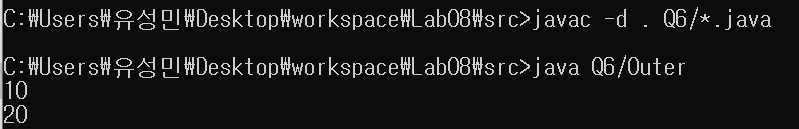
=> First at M2() method I made inner class’s object and invoked inner class’s method M1() like above code. And then at the static main method I made outer class’s object to invoke M2() method of Outer class. Because M2() method is non-static method, I made instance of Outer class at static main method to access M2() method. (Because static method can’t invoke(access) non-static method directly).

**Q5) The following code is a partial code. Complete the missing code to get the output.**

=> Main method of TestClass class is static. And M1 method is non-static method of inner class of Outer class. First as we know, static method can’t call non-static method directly. So I made Outer class’s object first and then made inner class’s object with outer class’s reference like above code. And then we can access M1 method at static main method.

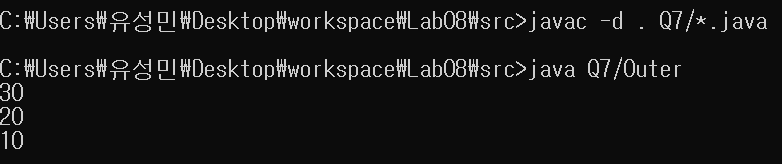
**Q6) Write and run the following code. From the output of the above code, a method (belongs to non-static inner class) can access fields of outer classes? Why?**

=> Because M1 is one of the field of Outer class, so it can access fields of Outer class. A method can access fields of Outer class

****

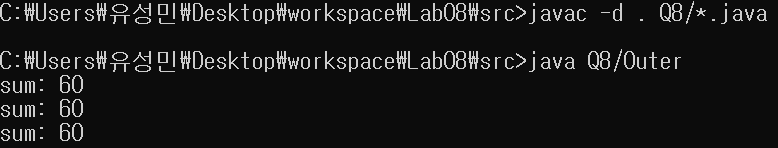
**Q7) The following code is a partial code. Complete the missing code to get the output.**

**=>** ‘x’ at first println sentence means the local variable of the method M1(), and ‘this.x’ means the local variable of the inner class. The last ‘Outer.this.x’ means the variable of the Outer class.

****At ‘static’ main method we can’t use non-static members directly. So I made Outer class’s object and inner class’s object to use the M1() method.

**Q8) Compile and run the following code. And answer the question.**

1. Can method M2(belongs to local inner class) access the local variable z (local to method M1)?
2. Explain your reason.

****=> Yes. M1 method has inner class and at that inner class, there is another method M2(). So inner class can use the local variable ‘z’ of M1 and the method M2(belongs to local inner class) can use the ‘z’.

**Q9) Compile and run the following code. And answer the question.**

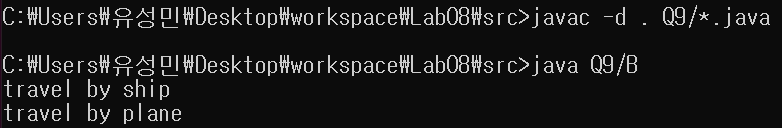
1. Why you get the above output?

=> The code ‘A a1 = new A(){};’ is form of anonymous inner class. ‘a1.M1()’ call the M1() method of the anonymous inner class. ‘a2’ is object of the ‘class A’, ‘a2.M1()’ call the M1() method at the class A.

1. After compiling, how many classes (.class files) are generated? List their names.

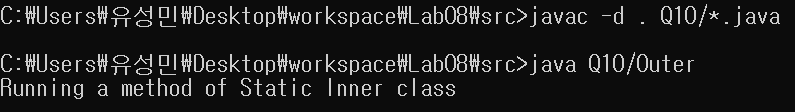
**스크린샷이(가) 표시된 사진

자동 생성된 설명**=> 3 .class files were made. A, B class file and B$1 class file. B$1.class means anonymous inner class. (anonymous inner class doesn’t have a class name)

****

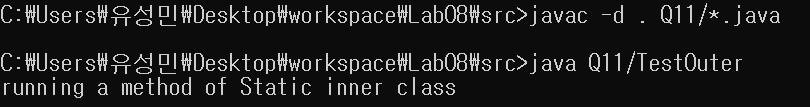
**Q10) The partial code and the output of the code is given below. Write the missing code.**

=> ‘StaticInner’ class is static. Also the main method is static. Between static, they can access directly. At static main method, if we want to use non-static M1() method of StaticInner class, we should make instance of the class and then we can use the method. So as you can see the code, I made the object of inner class, and used M1() method.

****

**Q11) The partial code and the output of the code is given below. Write the missing code.**

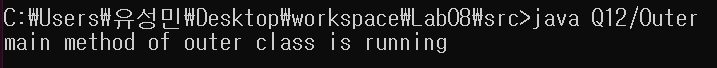
=> Because ‘StaticInner class’ is static, we don’t need to make Outer’s object at static main method. Because static can access static directly. So only making ‘Static Inner’ class’s object like above code, and then we can use non static method M1() with static class’s instance ‘si’.

****

**Q12) After writing the following code, answer the questions that follows.**

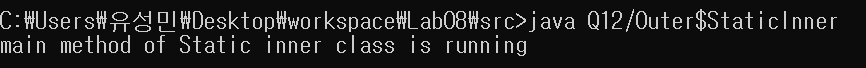
1. After compiling, run the code by using the following command.

java outer. What is the output?

=> When run as java outer, the main method of ‘Outer class’ is invoked and executed.

1. After compiling, run the code by using the following command.

java Outer$StaticInner. What is the output?

**=>** When run as java Outer$StaticInner, the main method of ‘StaticInner class’ which is inner class of the Outer class was invoked and executed.